Statistically robust 2D visual servoing

Andrew Comport, E. Marchand, François Chaumette

▶ **To cite this version:**

Andrew Comport, E. Marchand, François Chaumette. Statistically robust 2D visual servoing. IEEE Transactions on Robotics, 2006, 22 (2), pp.415-421. inria-00350284

HAL Id: inria-00350284
https://inria.hal.science/inria-00350284

Submitted on 6 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Statistically robust 2D visual servoing

Andrew I. Comport, Éric Marchand, François Chaumette

***Abstract*— A fundamental step towards broadening the use of real world image-based visual servoing is to deal with the important issue of reliability and robustness. In order to address this issue, a closed loop control law is proposed that simultaneously accomplishes a visual servoing task and is robust to a general class of image processing errors. This is achieved with the application of widely accepted statistical techniques such as robust M-estimation and LMedS. Experimental results are presented which demonstrate visual servoing tasks that resist severe outlier contamination.**

***Index Terms*— Visual servoing, Robust control law, M-estimators, LMedS.**

I. Introduction

VISUAL servoing is known to be a very efficient method for positioning and target tracking tasks [1]. However, its efficiency relies on correspondences between the position of tracked visual features in the current image and their position in the desired image which define a set of errors to be minimized. If these correspondences contain errors then visual servoing usually fails or converges upon a wrong position.

Overcoming these errors is often achieved by improving the quality of tracking algorithms [2], [3], [4] and feature selection methods [5]. This class of methods uses information measurements which are not directly related to the set of errors but based on external cues such as color and global motion estimation. These approaches provide a robust input estimate to the control loop, and as such treats outlier rejection in the image processing step, prior to the control step (see Fig. 1a). Considering redundant features [6] is also a simple way to improve positionning accuracy and reduce the sensitivity to noise. However it cannot allow to suppress completely the final positionning error in the presence of erroneous data.

Alternatively, the method proposed in this paper is based on a well founded and efficient formalism which directly uses the feature error vector to compute a statistical measure of confidence *at the control law level* (see Fig. 1b). In related literature, many different approaches exist to treat external sources of error. Amongst the robust outlier rejection algorithms [7], methods in computer vision have included the Hough Transform, RANSAC [8] and statistical methods such as Least Median of Squares (LMedS) [9] and M-estimators [10]. In this paper robust M-estimators [10] are employed because their formulation in terms of an Iteratively Re-weighted Least Square (IRLS) method allows its efficient integration directly into the control law. Weights that reflect the confidence in each visual feature location are computed and used in the control scheme. For the very first iteration of the control scheme, a more conservative robust estimator (LMedS [9]) is used and provides a more efficient rejection process when the errors due to outliers are of the same order of magnitude as the good measures.

Our approach features three main advantages. With respect to more classical robust visual servoing techniques that rely on a robust extraction of the visual features (Figure 1a), it bypasses intermediary decision steps which usually require thresholds to be tuned for each specific application. Second, the confidence in each visual feature relies on the value of all the other features. Finally, the computed uncertainty values do not act as a binary weight which completely rejects or accepts the feature. Each feature may either gain or loose certainty over time and during the execution of the control law.

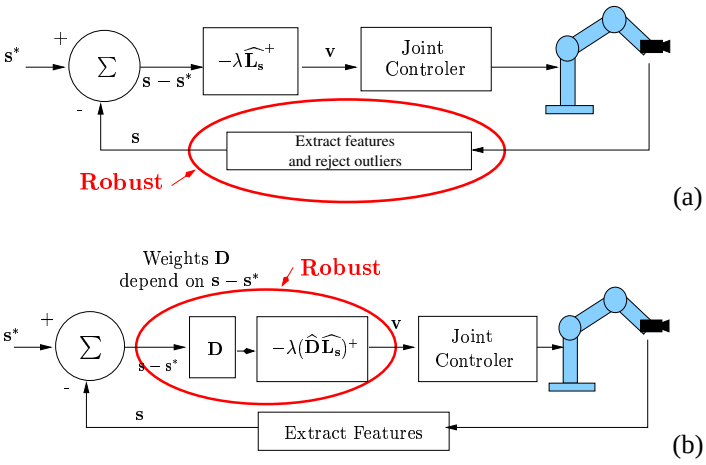


Fig. 1. (a) Traditional outliers rejection, (b) New proposed control law (see Section II-B for details)

Following an introduction to the method, the robust control scheme based on the M-estimators is detailed in Section II-B. In Section II-C, a method to initialize the weights based on LMedS is presented and finally in Section II-D we show how to combine both techniques. Experimental results are presented in Section III.

II. Robust Visual Servoing

A. Overview and motivations

The goal of visual servoing is essentially to minimize the error Δ between a set of visual features $\mathbf{s}(\mathbf{r})$, that depends of

Manuscript received April 25, 2005; revised September 28, 2005. This paper was recommended for publication by Associate Editor B. Nelson and Editor S. Hutchinson upon evaluation of the reviewers' comments. This paper was presented in part at the IEEE International Conference on Intelligent Robots and Systems, Las Vegas, NV, October 2003, and in part at the IEEE International Conference on Robotics and Automation, New Orleans, LA, April 2004.

The authors are with IRISA-INRIA Rennes, 35042 Rennes Cedex, France (e-mail: Eric.Marchand@irisa.fr; chaumett@irisa.fr).

Digital Object Identifier 10.1109/TRO.2006.870666

the camera pose $\mathbf{r}$, and a set of desired visual features $\mathbf{s}^*$:

$$\Delta = \sum_{i=1}^{k} \left(s_i(\mathbf{r}) - s_i^*\right)^2. \quad (1)$$

where k is the number of visual features in $\mathbf{s}$. Considering that $\mathbf{s}(\mathbf{r})$ is computed (from the image) with a sufficient accuracy is an important assumption. The control law that performs Δ minimization is usually handled using a least square approach [11], [1]. However when the data contains outliers, such a classical approach is no longer efficient. A solution to handle this problem is to perform a robust minimization. M-estimators can be considered as a more general form of Maximum Likelihood Estimators [10] because they permit the use of different minimization functions not necessarily corresponding to normally distributed data. Many functions have been presented in the literature which allow uncertain measures to be less likely considered and in some cases completely rejected [10], [9]. In the following, ρ is the objective function considered. The metric function to be minimized is modified to reduce the sensitivity to outliers. The new error to be minimized is then given by:

$$\Delta_{\mathcal{R}} = \sum_{i=1}^{k} \rho\Big(s_i(\mathbf{r}) - s_i^*\Big) \quad (2)$$

where $\rho(u)$ is a robust function that grows sub-quadratically and is monotonically nondecreasing with increasing $|u|$ [10].

B. Robust Control Law

Classical visual servoing control scheme have the following forms [11], [1]:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}_\mathbf{s}}^{+} (\mathbf{s} - \mathbf{s}^*) \quad (3)$$

where $\mathbf{v}$ is the camera velocity sent to the low-level robot controller and $\widehat{\mathbf{L}_\mathbf{s}}^{+}$ is the pseudo-inverse of a model or an approximation of the interaction matrix related to $\mathbf{s}$ (defined so that $\dot{\mathbf{s}} = \mathbf{L}_\mathbf{s}\mathbf{v}$) [11].

To embed a robust minimization in visual servoing, a modification of the control law is required to allow outlier rejection. For that, a weight is associated to each feature to specify a confidence in its location. This leads to the following new control law (see Fig 1b and [12]):

$$\mathbf{v} = -\lambda (\mathbf{D}\widehat{\mathbf{L}_\mathbf{s}})^{+} \mathbf{D}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big), \quad (4)$$

where $\mathbf{D}$ is a diagonal weighting matrix given by

$$\mathbf{D} = diag(w_1, \ldots, w_k)$$

As for all image-based visual servoing that use redundant features, it is unfortunately impossible to demonstrate the global asymptotic stability of the system. As demonstrated in [12], it is possible to demonstrate the local asymptotic stability when the outliers are assumed to be correctly detected, as soon as a sufficient number of features are not rejected so that $\mathbf{DL_s}$ is always of full rank (6 to control the 6 dof of the robot).

We now describe how the weights w_i are computed. In classical robust estimation algorithm, they are given by [10]:

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma} \quad (5)$$

where $\psi(u) = \frac{\partial \rho(u)}{\partial u}$ (ψ is the M-estimate and is also called the influence function), δ_i is the normalized residue given by

$$\delta_i = s_i - s_i^* - \text{med}\,(\mathbf{s} - \mathbf{s}^*)$$

where med $(\mathbf{s} - \mathbf{s}^*)$ corresponds to the median value taken across all the residues, and where σ is a scale that corresponds to a robust estimate of the standard deviation of the inlier data. It will be explained in full details later.

Of the various influence functions that exist in the literature we consider Tukey's hard re-descending function [10]. Tukey's function completely rejects outliers and gives them a zero weight (see Figure 2). This is of interest in visual servoing so that detected outliers have no effect on the robot motion. Its corresponding influence function is given by:

$$\psi(u) = \begin{cases} u(b^2 - u^2)^2 & , \text{ if } |u| \leq b \\ 0 & , \text{else} \end{cases} \quad (6)$$

where the proportionality factor for Tukey's function is $b = 4.6851$ which represents 95% efficiency in the case of Gaussian noise [13].

Typically, if the error is similar for most features and different for other features, the first ones will be considered as inliers (δ_i will be small and w_i near one) while the other ones will be considered as outliers (since δ_i will be large with w_i near zero). For very particular geometrical configuration this statistical method may not be efficient, but in most cases, it allows to reject correctly erroneous data.

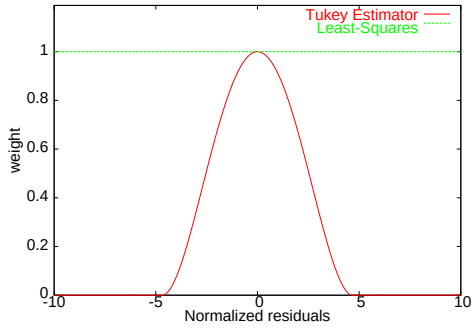


Fig. 2. Weight computed using Tukey's function for $\sigma = 1$ versus simple least-square method (where $w = 1$)

As already stated, the scale σ which appears in (5) is a robust estimate of the standard deviation of the good data. It is at the heart of the robustness of the function since, in visual servoing, this scale can vary dramatically during convergence. For voting methods and traditional M-estimators, scale has usually been treated as a tuning constant which can be chosen manually for a specific application. Alternatively, a robust statistic can be used to calculate it. One robust statistic, that allows to reject up to 50% of outliers, is the Median Absolute

Deviation (MAD) given by:

$$\widehat{\sigma} = \frac{1}{\Phi^{-1}(0.75)} \underset{i}{\text{med}} \left(|\delta_i - \underset{j}{\text{med}} \left(\delta_j\right)|\right). \tag{7}$$

where $\Phi()$ is the cumulative normal distribution function and $\frac{1}{\Phi^{-1}(0.75)} = 1.48$ represents one standard deviation of the normal distribution and is used to make the MAD consistent with the normal distribution. To date, a convergence proof for non-linear regression using the MAD only exists if it is calculated once as an auxiliary scale estimate due to the median's lack of asymptotic properties [13]. However, although convergence has not yet been proved due to discontinuities introduced by the median, experiments show that recomputing the MAD at each iteration gives better results. Other measures of the scale exist that reject a higher percentage of outliers than the MAD or simple median (see for example [14] or [15] when small datasets are considered). These methods do not, however, address the issue of computational efficiency which is of paramount importance in real-time frame-rate applications. The MAD is then a very good tradeoff between outliers rejection efficiency and computation efficiency.

C. Weights initialization using the LMedS approach

The weights w_i are computed from a statistic that depends only on the error $\mathbf{s}(\mathbf{r}) - \mathbf{s}^*$. When this error is large, it is difficult to detect outliers. Indeed, the error $\Delta_i = s_i + \epsilon_i - s_i^*$ (where ϵ_i is an "aberration" due to imprecision in data extraction) may not be statistically significant wrt. the other errors. If some outliers are not detected as such, corresponding weights are not equal to zero and the robot trajectory can be strongly perturbed. Therefore it is important to detect the features that are likely to be outliers prior the beginning of the servo process and to initialize adequately the weights. This can be achieved using the LMedS robust estimator.

The LMedS [9] method estimates the parameters by solving the non-linear minimization problem:

$$\min_{\mathbf{x}} \underset{i=1,\ldots k}{\text{med}} \; r_i^2$$

where r_i is the residual for each available measure and k is the number of measures. In our case, we are looking for the solution $\mathbf{x}$ of the linear system $\widehat{\mathbf{L}}_{\mathbf{s}}\mathbf{x} = (\mathbf{s} - \mathbf{s}^*)$, and the residual r_i is given by:

$$r_i^2 = \left(\widehat{\mathbf{L}}_{s_i}\mathbf{x} - (s_i - s_i^*)\right)^2$$

where $\widehat{\mathbf{L}}_{s_i}$, s_i and s_i^* are, respectively, the i-th line of the interaction matrix $\widehat{\mathbf{L}}_{\mathbf{s}}$ and the i-th component of vectors $\mathbf{s}$ and $\mathbf{s}^*$.

Unlike M-estimators, the LMedS method cannot be reduced to a weighted least squares problem. It must be solved by a search in the space of possible estimates generated from the data. This space is usually very large (and this is why we do not use it inside the control law but only for the initialization).

The algorithm described below enables a robust detection of the outliers within the whole set of features. Given k features $s_i, i = 1 \ldots k$:

1) draw N subsamples $\mathbf{s}_J, J = 1 \ldots N$ of n independent visual features. The maximum number of subsamples is $N_{\max} = \binom{k}{n}$ where n is the minimal number of features that allows to perform a positioning task ($6 \leq n \leq k$). Therefore if k is large, $N_{\max}$ may be very large and a Monte Carlo technique can be used to draw N subsamples ($N << N_{\max}$) that ensure a good outlier detection probability (see [9] for details).
2) for each subsample J, we compute the solution $\mathbf{x}_J$ according to:
$$\mathbf{x}_J = \widehat{\mathbf{L}}_{\mathbf{s}_J}^{+}(\mathbf{s}_J - \mathbf{s}_J^*)$$
3) For each $\mathbf{x}_J$, we determine the median of square residuals, denoted M_J, with respect to the whole set of features, that is:
$$M_J = \underset{i=1\ldots k}{\text{med}} \left(\widehat{\mathbf{L}}_{s_i}\mathbf{x}_J - (s_i - s_i^*)\right)^2$$
4) We retain the value M^* that is minimal among all N M_J's. The corresponding $\mathbf{x}_J$ could be also of interest to control the robot, but we prefer to use a weighted control as described in Section II-B for such purpose since only n features would be considered regardless of the number of inliers. M^* will now be used to detect outliers.

LMedS must be carefully designed to detect and remove outliers. Following the work of Rousseeuw [9], we assign a binary weight to each feature according to:

$$w_i = \begin{cases} 1 & \text{if } |r_i| \leq 2.5\widehat{\sigma} \\ 0 & \text{else} \end{cases} \tag{8}$$

where $\widehat{\sigma}$ is (as in (7)) a scale estimate defined by the robust statistic given by (see [9] for details):

$$\widehat{\sigma} = 1.4826(1 + 5/(k-n))\sqrt{M^*} \tag{9}$$

where $1 + 5/(k-n)$ is a small-sample correction factor which makes the scale unbiased when a small dataset is considered [9], [15], [14].

D. Full weights computation

Though not very complex, the LMedS-based outliers rejection algorithm requires some processing time that is not yet compatible with a 25Hz loop. That is why we consider it only for initialization. If we denote w^{tukey} the weights computed from (5) and w^{LMedS} the weights obtained from (8), a global weight can be defined as:

$$w_i = (1 - \alpha)w_i^{tukey} + \alpha w_i^{LMedS} \tag{10}$$

where $\alpha = 1 - exp(-\beta_1 \|\mathbf{D}_{(t)}(\mathbf{s}_{(t)} - \mathbf{s}^*)\|)$. When $\mathbf{D}_{(t)}(\mathbf{s}_{(t)} - \mathbf{s}^*)$ is large, α is close to 1 and w^{LMedS} is mainly considered. However, since LMedS are conservative, features considered as outliers by this method may be in fact inliers. Considering a combination between LMedS and M-estimators allows one to reconsider the LMedS initialization during the servo. Similarly, if a feature has been considered as an inlier, the M-Estimator influence increases in order to consider that the feature may become an outlier during the task execution (note that α decreases since $\mathbf{D}_{(t)}(\mathbf{s}_{(t)} - \mathbf{s}^*)$ decreases progressively).

Furthermore, to smooth the weight evolution (and then the camera trajectory) it is also possible to introduce a memory

process. The first possible solution is to smooth the weights evolution such that :

$$w_i'(t) = \beta_2 w_i(t) + (1-\beta_2) w_i(t-1) \tag{11}$$

with $\beta_2 \in [0:1]$ has to be tuned. The second solution is to smooth in a similar way the median computed in (7). Indeed, noise in the weights computation is mainly due to instability in the median computation due to the small number of data. Such a problem does not arise when the number of data k increases. In the presented results, the later solution has been retained and we have set $\beta_2 = 0.2$.

III. Experimental results

The complete implementation of robust visual servoing, including tracking and control, was carried out on an experimental test-bed involving a CCD camera mounted on the end effector of a six degrees-of-freedom robot. We have considered a positioning task. From an initial position, the robot has to reach a desired position expressed as a desired position of the object in the image.

A. Visual features and weights computation.

In these experiments visual features are given as a set of point coordinates extracted from the image. If n points are considered, $\mathbf{s}$ is a vector defined as $\mathbf{s} = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$ where (x_i, y_i) are the coordinates of the $i-th$ point. Interaction matrix $\mathbf{L_s}$ is a $2n\times$ 6 matrix given by $\mathbf{L_s} = (\mathbf{L_s}_1, \ldots, \mathbf{L_s}_n)$ with:

$$\mathbf{L_s}_i = \begin{bmatrix} -\frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_i y_i & -(1+x_i^2) & y_i \\ 0 & -\frac{1}{Z_i} & \frac{y_i}{Z_i} & 1+y_i^2 & -x_i y_i & -x_i \end{bmatrix}$$

Weights are computed considering the two residual vectors:

$$\forall k = 1 \ldots n, \quad \begin{cases} \Delta_{2k} = x_k - x_{k_d} \\ \Delta_{2k+1} = y_k - y_{k_d} \end{cases}$$

Since weights w_{2k} and w_{2k+1} reflect the confidence we have in the same point, we define elements of the weights matrix $\mathbf{D}$ as $\mathbf{D}_{2k,2k} = \mathbf{D}_{2k+1,2k+1} = \min(w_{2k}, w_{2k+1})$.

B. Experiments with dots

In the first experiment a pattern made with twelve white dots is considered. Tracking such a simple pattern allows to validate the efficiency of the new control law.

1) Effect of large errors: In this first set of experiments, four cases are compared:

- **[Exp 1]** a reference experiment with no error and using the control law given in (4) with $\mathbf{D} = \mathbf{I}$;
- **[Exp 2]** an experiment with the same control law ($\mathbf{D} = \mathbf{I}$) but artificial noise has been added in data extraction: a large error (more than 60 pixels at convergence) was introduced into the extracted coordinates of two points which were voluntarily inverted ;
- **[Exp 3]** this experiment is similar to [Exp 2] but weights are computed as described by equation (5) using the Tukey M-estimator ;
- **[Exp 4]** in this experiment we extend [Exp 3] by initializing weights using the LMedS method presented in section II-C ;

As expected, the classical visual servoing converges successfully toward the desired position when no error is introduced. A large error on two points [Exp 2] implies the convergence of the control law toward a position that is not the desired one (see Table 1 and Fig. 3c). The distance between the *outliers* points is of 38 pixels in the initial image and of 68 pixels at the end of the positioning task. An error on the final coordinates of each inlier point can also be observed. Let us note that, in some cases, a complete divergence of the control law can even be observed (which means that the global error $\|\mathbf{s} - \mathbf{s}^*\|$ along with the camera velocity increases and the robot moves rapidly toward its joint-limits).

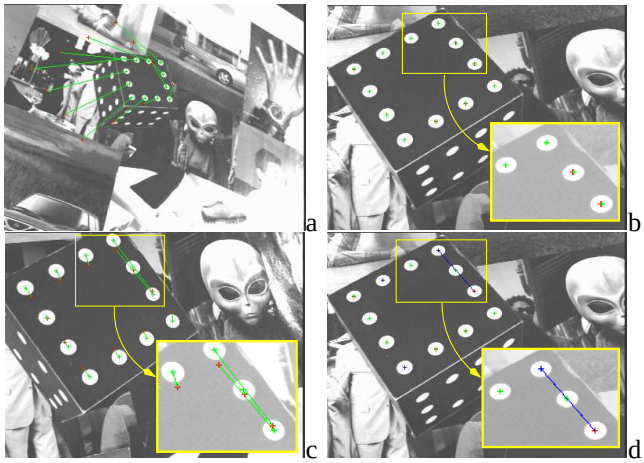


Fig. 3. Result of a positioning task using classical and robust control law. Image (a) shows the initial camera position, image (b) shows the final camera position for the reference experiment, image (c) shows the final camera view when no robust control law is considered (an error may be observed) and image (d) shows the final camera view when the robust control law is considered.

Our new method improves the behavior of the positioning task. Indeed in the two other experiments where a weighting matrix is introduced in the control law [Exp 3-4], the camera reaches its desired position with a very good accuracy despite the errors introduced in the data.

However when only M-estimation is considered, the outliers are not detected at the beginning of the task (see weights evolution on Fig. 4) since their error is not statistically significant wrt. the other errors. This leads to disturbances in the camera trajectory which is different from the "reference" trajectory (see Fig. 5). When the more conservative LMedS-based weight initialization process is considered, it is possible to detect outliers before the beginning of the positioning task.

2) Effect of small errors: In the next experiments [Exp 6-7-8] a small error is added to the extracted position of four dots. A partial occlusion is made by sticking a rubber on the target, so that it does occur equally during the whole experiment, from the initial image to the final image. Since this occlusion of the dots (around 60%) does not exist in the reference image, it adds a small error (5 pixels) onto the position of the center of gravity of these features (see Fig. 6). Even though the

Position			Tx	Ty	Tz	Rx	Ry	Rz
no error	reference	[Exp 1]	-0,1	-0,1	0,1	-0,01	-0,02	0,07
large errors	no robust	[Exp 2]	-127,3	-102,9	29,9	-3,14	-15,63	22,70
	Mest Tukey	[Exp 3]	-0,1	-0,1	-0,1	-0,02	-0,03	0,12
	LMedS+Mest	[Exp 4]	-0,2	0,1	-0,2	-0,02	-0,07	0,07
small errors	no robust	[Exp 6]	-35,6	-12,8	3,0	-1,42	-1,43	6,47
	LMedS+Mest	[Exp 7]	-0,4	-0,5	-0,3	-0,04	-0,11	0,30
	LMedS+Mest (no MAD)	[Exp 8]	11,9	6,2	9,7	0,61	1,87	-2,19

TABLE I

POSITIONING ACCURACY: EACH LINE DISPLAYS THE ERROR (IN MM AND DG) BETWEEN THE DESIRED AND FINAL CAMERA POSITION.

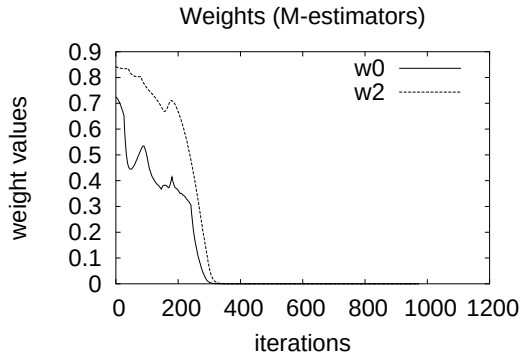


Fig. 4. Weight evolution for the two points detected as outliers using only M-estimators.

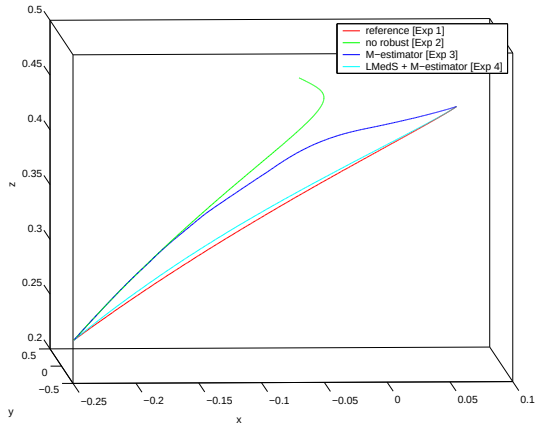


Fig. 5. Camera trajectory for large matching errors.

error is quite small, the positioning errors are significant using classical visual servoing (see Fig. 6.b and Table I [Exp 6] wrt [Exp 1]).

Even with the conservative LMedS estimator, the small errors cannot be detected at the very first iteration. When the standard deviation of the noise measure (MAD) $\widehat{\sigma}$ is recomputed at each iteration of the control law [Exp 7], after several iterations (that is when δ has decreased and therefore when the effects of the outliers are more visible) the outliers are detected and the positioning task is achieved with a very good accuracy. However, if the MAD is not recomputed at each iteration [Exp 8], since the outliers are very difficult to detect from the first computation of the error δ, the initial computed MAD is not adapted and robot converges to a wrong position. These experiments underlines the fact that recomputing the MAD at each iteration ensures a far better behavior of the control law. Although in this experiment, the outliers were present at the beginning of the positioning task, this approach is also efficient for transient outliers that appear during the task, as can be seen in the next experiment.

C. Experiments with SSD trackers.

In the last experiment we have considered far more complex textured images (see Fig. 7) acquired by a low quality camera that provides poor quality image. In such condition tracking features is difficult and is not very reliable. The chosen algorithm is based on a classical SSD algorithm (Shi-Tomasi-Kanade). We defined the position to reach with a reference image (see Fig. 7b). Points of interest are extracted (using the Harris detector) and are matched with similar points, extracted from the image acquired from the initial camera location, using the Image-Matching software [16]. Therefore at the beginning of the task there is no matching error (as detected by the LMedS at initialization since all the weights are set to 1).

Firstly, it can be noted that with the use of a classical control law and due to excessive mis-tracking, the camera was not able to reach the desired position. Fig. 7c shows the difference between the desired image (Fig. 7b) and the last one acquired by the camera (Fig. 7d). In Fig. 7d and 7f, red crosses are the initial points location, blue crosses are their desired locations while the green crosses are the final points location. Point trajectories are in red (60 points are tracked). Next the robust control law was applied. The desired position was obtained with good accuracy (less that 7 mm in translation) even with very poor experimental conditions as can be seen from the difference (see Fig. 7e) between the desired image (Fig. 7b) and the last one acquired by the camera (see Fig. 7f). Fig. 7f shows that many points have been detected has outliers during the task.

IV. CONCLUSION

Previous visual servoing methods have only considered outlier rejection in the image processing step. In this paper a novel visual servoing method has been proposed that rejects errors in feature extraction, tracking and matching at the control law level. Experimental results show the efficiency of the approach for a positioning task on case-study examples

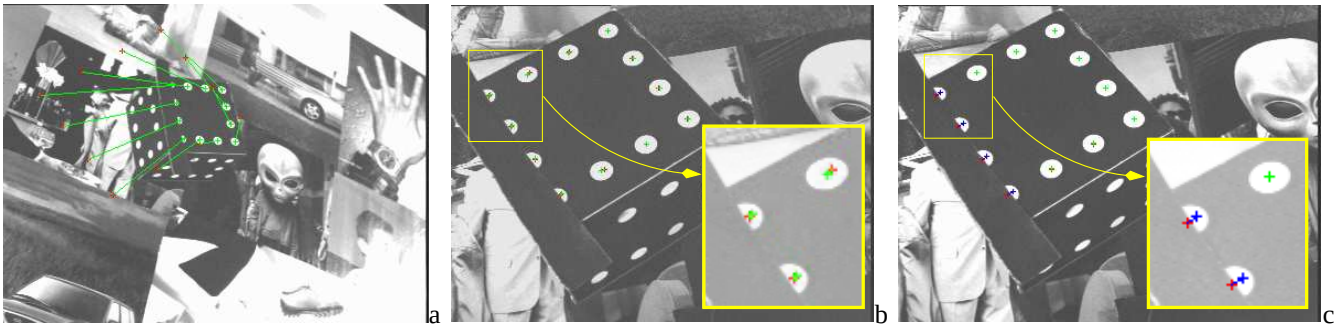
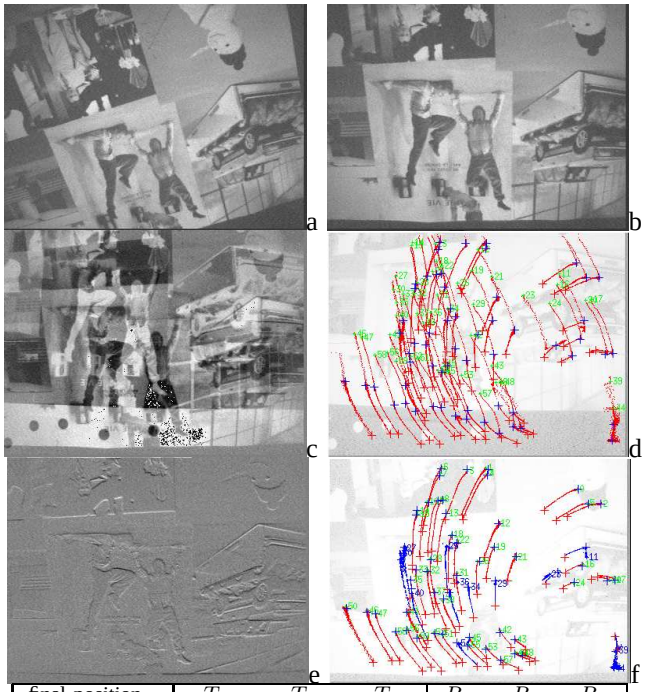


Fig. 6. Small errors on many points (a "row" of points is partially hidden): (a) Initial image, (b) Final image with classical control law [Exp 6] (c) Final image with robust estimation [Exp 7]



final position	T_x	T_y	T_z	R_x	R_y	R_z
Desired	-131.2	-17.6	259.2	-0.1	19.2	-0.5
Without robust	-257.6	-356.7	295.8	-0.9	-5.1	8.0
Robust	-137.3	-16.4	258.6	-0.1	19.2	0.1

Fig. 7. Visual servoing based on the tracking of points of interest: (a) initial image, (b) desired image, (c-d) classical visual servoing control law (difference image and point trajectory on the final image), (g-h) same with robust visual servoing, (i) positioning errors (mm and dg)

and on real images. In all cases a great improvement in the positioning accuracy has been observed wrt. a non robust control law.

REFERENCES

[1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[2] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, "Making good features track better," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Santa Barbara, USA, June 1998, pp. 178–183.

[3] D. Kragic and H. Christensen, "Cue integration for visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 1, pp. 19–26, Feb. 2001.

[4] A.I. Comport, E. Marchand, and F. Chaumette, "A real-time tracker for markerless augmented reality," in *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, Tokyo, Japan, Oct. 2003, pp. 36–45.

[5] N. P. Papanikolopoulos and P. K Khosla, "Selection of features and evaluation of visual measurements for 3D robotic visual tracking," *Int. Symp. on Intelligent Control.*, pp. 320–325, Aug. 1993.

[6] K. Hashimoto, A. Aoki, and T. Noritsugu, "Visual servoing with redundant features,," in *Proc. of 35th IEEE Conf. on Decision and Control*, Kobe, Japan, Dec. 1996, pp. 2482–2484.

[7] C.-V. Stewart, "Robust parameter estimation in computer vision," *SIAM Review*, vol. 41, no. 3, pp. 513–537, Sept. 1999.

[8] N. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," *Communication of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[9] P.J. Rousseeuw and A.M. Leroy, *Robust Regression and Outlier Detection*, John Wiley and Sons, New York, 1987.

[10] P.-J. Huber, *Robust Statistics*, Wiler, New York, 1981.

[11] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, June 1992.

[12] A.I. Comport, M. Pressigout, E. Marchand, and F. Chaumette, "A visual servoing control law that is robust to image outliers," in *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'03*, Las Vegas, Nevada, Oct. 2003, vol. 1, pp. 492–497.

[13] P.-W. Holland and R.-E. Welsch, "Robust regression using iteratively reweighted least-squares," *Comm. Statist. Theory Methods*, vol. A6, pp. 813–827, 1977.

[14] H. Wang and D. Suter, "Robust adaptive-scale parametric estimation for computer vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1459–1474, Nov. 2004.

[15] P.J. Rousseeuw and S Verboven, "Robust estimation in very small samples," *Computational Statistics and Data Analysis*, vol. 40, pp. 741–758, 2002.

[16] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence*, vol. 78, pp. 87–119, Oct. 1995.